Some Hints

Hints

▶ The exam will potentially cover everything in the course so far, but will focus on material we've seen since the first midterm.
▶ There will be a question that is also a one-star exercise from the book.
▶ There will be a question similar to problem 6 from midterm 1 ("Which properties remain true if we change one of the type systems we’ve studied in the following way...?")
▶ There will be (at least) one question based on one of the proofs in chapter 15.
▶ For PhD students, there will be a question involving subtyping and references.

Review

What are the types of these expressions?
▶ λx:Bool→Bool. x (x (x (true)))

What are the types of these expressions?
▶ λx:Bool→Bool. x (x (x (true)))
▶ (λx:Bool. λy:Bool→Bool. true) false (λz:Bool→Bool. true)
What are the types of these expressions?

- \( \lambda x : \text{Bool} \rightarrow \text{Bool}. x (x (x (\text{true}))) \)
- \((\lambda x : \text{Bool}. \lambda y : \text{Bool} \rightarrow \text{Bool}. \text{true})\)
- \(\text{false (\lambda z : \text{Bool} \rightarrow \text{Bool}. \text{true})}\)
- \((\lambda x : \text{Bool}. \lambda y : \text{Bool}. \text{error})\)
- \(\text{false false false false false}\)
- \((\lambda x : \text{Bool}. \lambda y : \text{Bool}. \text{true})\)
- \(\text{false false false false false}\)
- \(\text{try (if (\lambda x : \text{Bool}. x) \text{error then (error false) else error) with (\lambda y : \text{Bool} \rightarrow \text{Bool}. y)}\)

For reference: Typing rules for exceptions

\[
\Gamma \vdash \text{error} : T \quad \text{(T-Error)}
\]
\[
\Gamma \vdash t_1 : T \quad \Gamma \vdash t_2 : T \quad \Gamma \vdash \text{try } t_1 \text{ with } t_2 : T \quad \text{(T-Try)}
\]

Give the result of evaluation and the final store after each of these expressions is evaluated to a normal form starting in the empty store.

- \(\text{let } x = \text{ref } 0 \text{ in } \text{let } y = \text{ref } 1 \text{ in } \text{y} := 3 \text{ !x}\)
- \(\text{let } x = \text{ref } 0 \text{ in } \text{let } y = \text{ref } 1 \text{ in } \text{y} := 3 \text{ !x}\)
- \(\text{let } x = \text{ref } 0 \text{ in } \text{let } y = \text{ref } 1 \text{ in } \text{let } x = y \text{ in } \text{!x}\)
Give the result of evaluation and the final store after each of these expressions is evaluated to a normal form starting in the empty store.

1. `let x = ref 0 in
   let y = ref 1 in
   y := 3
   !x`

2. `let x = ref 0 in
   let y = ref 1 in
   let x = y in !x`

3. `let x = ref 0 in
   let y = x in
   let x = y in !x`

For reference: Evaluation rules for references

\[
\frac{l \notin \text{dom}(\mu)}{\text{ref } v_1 | \mu \rightarrow l | (\mu, l \mapsto v_1)} \quad \text{(E-RefV)}
\]

\[
\frac{\mu(l) = v}{!l | \mu \rightarrow v | \mu} \quad \text{(E-DerefLoc)}
\]

\[
\frac{l := v_2 | \mu \rightarrow \text{unit} | [l \mapsto v_2]|\mu}{(E-Assign)}
\]

(Plus several congruence rules.)

Which of the following functions could evaluate to 42 when applied to a single argument and evaluated with a store of the appropriate type?

1. `\lambda x: \text{Ref Nat.} \; !x+1`

2. `\lambda x: \text{Ref Nat.} \; x`

Which of the following functions could evaluate to 42 when applied to a single argument and evaluated with a store of the appropriate type?

1. `\lambda x: \text{Ref Nat.} \; !x+1`

2. `\lambda x: \text{Ref Nat.} \; x`

3. `\lambda x: \text{Ref Nat.} \; (x:=3; l_1:=42; !l_1)`
Preservation and progress for chapter 13

- The preservation and progress proofs for \( \lambda \rightarrow \) are just sketched in TAPL.
- Working out the details for yourself is an excellent exercise.
- A question based on this proof may appear on the final exam, but will not appear on the coming midterm.

Subtyping

For each of the following pairs of terms, say whether the one on the left is a subtype of the one on the right, a supertype, equivalent, or incomparable.

- \{\} \rightarrow \{} \rightarrow \text{Top} and \text{Top} \rightarrow \text{Top}
- \text{Top} \rightarrow \text{Top} \rightarrow \{} \rightarrow \{} and \text{Top} \rightarrow \{} \rightarrow \text{Top}
- \{a: \text{Top}, b: \{c: \text{Top}, d: \text{Top}\}\} and \{b: \{d: \text{Top}, c: \text{Top}\}, a: \text{Top}\}
- <l: \text{Top}, m: \{n: \text{Top}\}> and <m: \{n: \text{Top}, o: \text{Top}\}>
- <> \rightarrow \text{Top} and {} \rightarrow \text{Top}
Subtyping

Draw a subtyping derivation for the following statement:

\[(\text{Top} \rightarrow \{x: \text{Nat}\} \rightarrow \{x, y: \text{Nat}\}) \triangleleft (\{} \rightarrow \{\} \rightarrow \{y: \text{Nat}\}\)\]

For reference: Subtyping rules

\[
\begin{align*}
S &\triangleleft S & (\text{S-Refl}) \\
S &\triangleleft U & U \triangleleft T & (\text{S-Trans}) \\
S &\triangleleft T & & (\text{S-Trans}) \\
\{l_i : T_i \mid i \in 1 \ldots n\} &\triangleleft \{l_j : T_j \mid j \in 1 \ldots n\} & (\text{S-RcdPerm}) \\
\{l_i : S_i \mid i \in 1 \ldots n\} &\triangleleft \{l_j : T_j \mid j \in 1 \ldots n\} & (\text{S-RcdPerm}) \\
T_1 &\triangleleft S_1 & S_2 &\triangleleft T_2 & (\text{S-Arrow}) \\
S_1 \rightarrow S_2 &\triangleleft T_1 \rightarrow T_2 & (\text{S-Arrow}) \\
S &\triangleleft \text{Top} & (\text{S-Top})
\end{align*}
\]

Ascription as a derived form

- Someone asked to work exercise 11.4.1 part 2 today.
- But the solution is somewhat technical and would take too much time to discuss in detail.
- This exercise is not needed for the exam.

The Hints, again

- The exam will potentially cover everything in the course so far, but will focus on material we’ve seen since the first midterm.
- There will be a question that is also a one-star exercise from the book.
- There will be a question similar to problem 6 from midterm 1 (“Which properties remain true if we change one of the type systems we’ve studied in the following way...?”)
- There will be (at least) one question based on one of the proofs in chapter 15.
- For PhD students, there will be a question involving subtyping and references.